

Final Report

Self-Timed Digital System Design

Principal Investigator: Dr. Alvernon Walker

Sponsoring Agency: NASA Goddard

Contractor: North Carolina A&T State University

Contract No. :2905-98-0079

Period of work covered: May 16, 1998 - May 17, 2001

Abstract

A timing and control strategy that can be used to realize synchronous systems with a level of performance that approaches that of asynchronous circuits or systems was developed in this work. This approach is based upon a single-phase synchronous circuit/system architecture with a variable period clock. The handshaking signals required for asynchronous self-timed circuits are not needed. Dynamic power supply current monitoring is used to generate the timing information, that is comparable to the completion signal found in self-timed circuits; this timing information is used to modify the circuit clock period.

Table of Contents

Abstract	i
1.Introduction	1
2.DPSC	5
3.Example	7
4.Conclusion	9
Appendix A	10

1. Introduction

The throughput, i.e. the number of instructions that are completed by a processor per processor cycle, and the number of processor cycles between two instruction initiations, i.e. latency, is determined by the clock period (i.e. a processor cycle) of high-performance superscalar/vector processors that are implemented with pipelines where there is no feedback between pipeline stages, i.e. linear pipelines. This is also true for nonlinear pipelines although the relationship between processor cycles, throughput and latency is more complex. The period of the processor cycle is determined by the setup and hold time requirements of the interstage latches and the worst case propagation delay in the slowest pipeline stage. The worst case propagation delay of this stage is typically substantially larger than the setup and hold time of the interstage latches in scalar pipelines, i.e. pipelines with a small number of stages. It occurs when the critical path in the slow pipeline stage is active. This critical path is active for a subset of the inputs to this stage therefore processor performance is sacrificed if the processor cycle is generated with a fixed-period global clock, i.e. a *synchronous* processor realization. An adaptive timing/clocking strategy that could determine when the outputs of the slowest pipeline stage are valid would resolve this problem and lead to substantially higher processor performance. Two general design approaches are used to realize processor subsystems that indicate when their output is valid, hence realizing a digital system with adaptive timing. In one subsystem design approach event ordering and timing is based upon the local generation of timing information via the intrinsic propagation delays of active circuit paths, circuits designed with this timing and control approach are called *asynchronous* circuits. The second

approach is based upon a special class of asynchronous circuits called *self-timed circuits*, these circuits are modular and utilize handshaking, start and done signaling for intermodule communication and module process initiation and completion, where a module is a subfunction/function realization.

The main disadvantage of asynchronous systems is that complex circuits which are difficult to design are required to realize relatively simple operators/functions. The performance of self-timed circuits is comparable to that of asynchronous circuits without the associated complexity and design problems. Although self-timed circuits are not as complex as asynchronous circuits, they are far more complex than synchronous circuits due to the redundancy in the datapath (or circuit critical path) that is required to generate the completion or done signal, i.e. a signal that indicates the output of the module is valid [1]. The complexity of self-timed modules are also a result of the handshaking circuits that are required for intermodule communication. Here we propose a timing and control strategy that can be used to realize synchronous systems with performance that approaches that of asynchronous circuits or systems without the associated circuit complexity. This approach is based upon a single-phase synchronous circuit/system architecture with a variable period clock. The handshaking signals required for self-timed circuits are not needed because of this system architecture. Dynamic power supply current (DPSC) monitoring is used to generate the timing information, that is comparable to the completion signal found in self-timed circuits, which is used to modify the circuit clock period. This current is used to control the current sources in the ICO (i.e. current controlled oscillator) that generates the processor global clock.

The current supplied by the power and ground rail in static CMOS logic circuits can be

divided into two general classes: quiescent and dynamic. The steady-state current I_{DDQ} that flows from the power rail V_{DD} (or into the ground rail) for a given input vector $\langle V_i \rangle$ is called quiescent current. This current is approximately zero in a fault-free circuit [3,4]. The second general class of rail current,

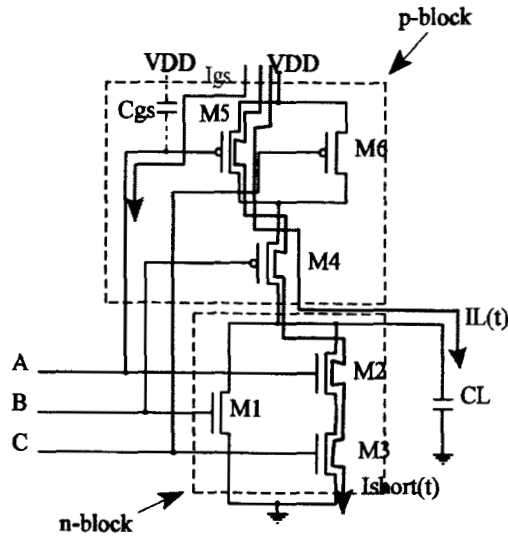


Figure 1 Complex Static CMOS Gate

DPSC $I_{DD}(t)$, is produced when a transition appears at the output of the gate attached to the power/ground rail for an input pair $\langle V_1, V_2 \rangle$. This current only consists of the rail currents of gates on the circuit critical path. The current supplied by the rail voltage V_{DD} during a low-to-high transition at the output of a static CMOS gate is called the p-block DPSC $I_{DDp}(t)$, where the p-block is an array of interconnected p-channel transistors and the n-block is a group of connected n-channel devices in the static CMOS gate as shown in Fig. 1. The n-block DPSC $I_{DDn}(t)$ is the ground rail current that is produced during a high-to-low transition at the gate output. This current consists of the following supply (or ground) currents: short-circuit current $I_{short}(t)$, load C_L displacement current

$I_L(t)$, gate input capacitance C_{gs} , displacement current $I_{cgs}(t)$ and block device leakage current $I_{leakage}(t)$ [5].

In order to fully realize the system level performance of this timing approach we focused our investigation on the physical design, circuits and processor architectural issues that are associated with the design and realization of a high-performance self-timed synchronous superscalar microprocessors. The primary focus of our research is in the following three areas:

- I. Development and design issues that are associated with Dynamic Power Supply Current Monitor (DPSCM) based self-timed synchronous functional unit pipeline stage and on-chip bus realizations. This includes the clock control hierarchy, circuit and physical design. It also includes datapath physical design bitwise scaling issues that will increase clock period controllability and average pipeline performance.
- II. Processor architectural and performance issues that are associated with a variable global/local processor cycle processor architecture. The issues included in this processor level investigation will be data and procedural dependency, resource conflicts, instruction parallelism and machine parallelism in a self-timed synchronous superscalar computer architecture. The main objective of this architectural and performance level research is to determine the optimal processor architecture and system/processor level clock control hierarchy.

- III. On-chip and off-chip memory/bus timing and performance issues associated with a variable processor cycle architecture. The goal of this work is to determine the optimal memory architecture/ interface realization for a self-timed synchronous superscalar processor.

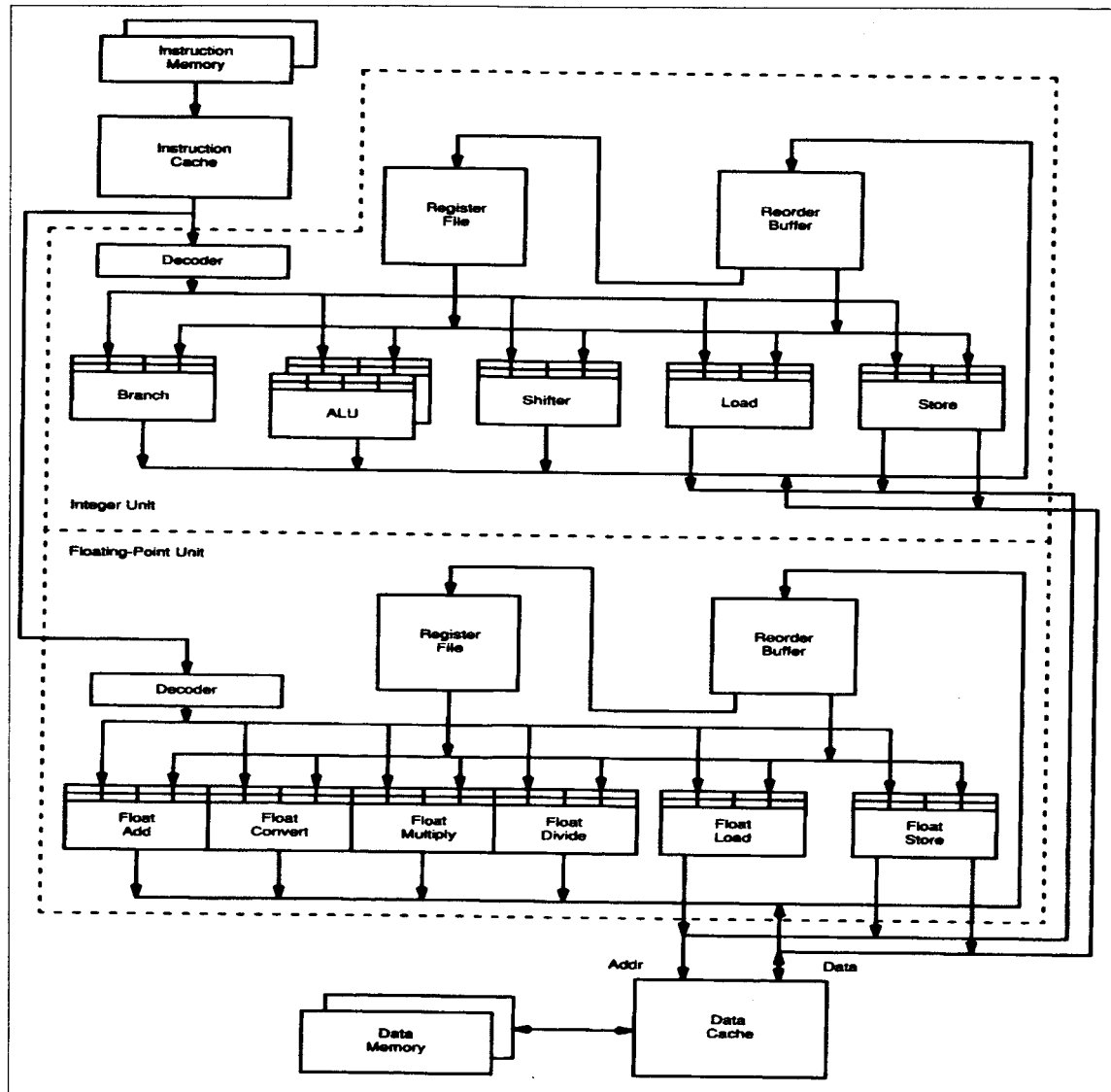


Figure 2 Basic Superscalar Architecture

The basic architecture of the self-timed/locally-timed superscalar processor under development in this work is shown in Fig. 2. A VHDL model and the physical design of the integer execution units (i.e. ALU and Shift Units in Fig. 2) were completed. The physical design was done in 2.0 micron bulk CMOS. During this work several binary adder structures were studied primarily because this structure contains the critical path of the integer execution unit. These structures include: the dynamic CMOS carry-lookahead structure proposed by Yun et. al., the transmission gate realization of the Manchester chain proposed by Abnous and Behzad, DCVS realization proposed by Renaudin and Hassan and the structure proposed by Compton and Albicki. A VHDL and a physical design (in 2.0 micron CMOS) of each of these proposed designs was done during this work to evaluate each of these approaches in an integrated circuit.

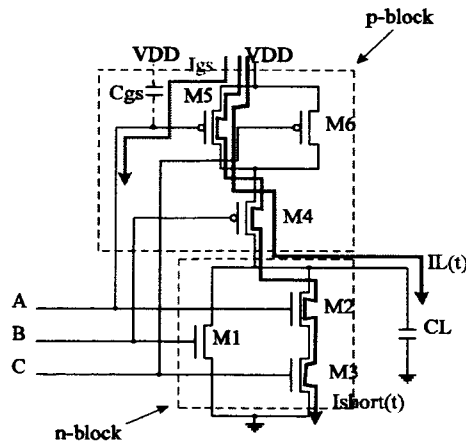


Figure 3 Complex Static CMOS Gate

Copies of the published work from this project are in appendix A.

2. DPSC

The current supplied by the power and ground rail in static CMOS logic circuits can be divided into two general classes: quiescent and dynamic. The steady-state current I_{DDQ} that flows from the power rail V_{DD} (or into the ground rail) for a given input vector $\langle V_1 \rangle$ is called quiescent current. This current is approximately zero in a fault-free circuit [3,4]. The second general class of rail current, DPSC $I_{DD}(t)$, is produced when a transition appears at the output of the gate attached to the power/ground rail for an input pair $\langle V_1, V_2 \rangle$. This current only consists of the rail currents of gates on the circuit critical path. The current supplied by the rail voltage V_{DD} during a low-to-high transition at the output of a static CMOS gate is called the p-block DPSC $I_{DDp}(t)$, where the p-block is an array of interconnected p-channel transistors and the n-block is a group of connected n-channel devices in the static CMOS gate as shown in Fig. 2. The n-block DPSC $I_{DDn}(t)$ is the ground rail current that is produced during a high-to-low transition at the gate output. This current consists of the following supply (or ground) currents: short-circuit current $I_{short}(t)$, load C_L displacement current $I_L(t)$, gate input capacitance C_{gs} displacement current $I_{cgs}(t)$ and block device leakage current $I_{leakage}(t)$ [5].

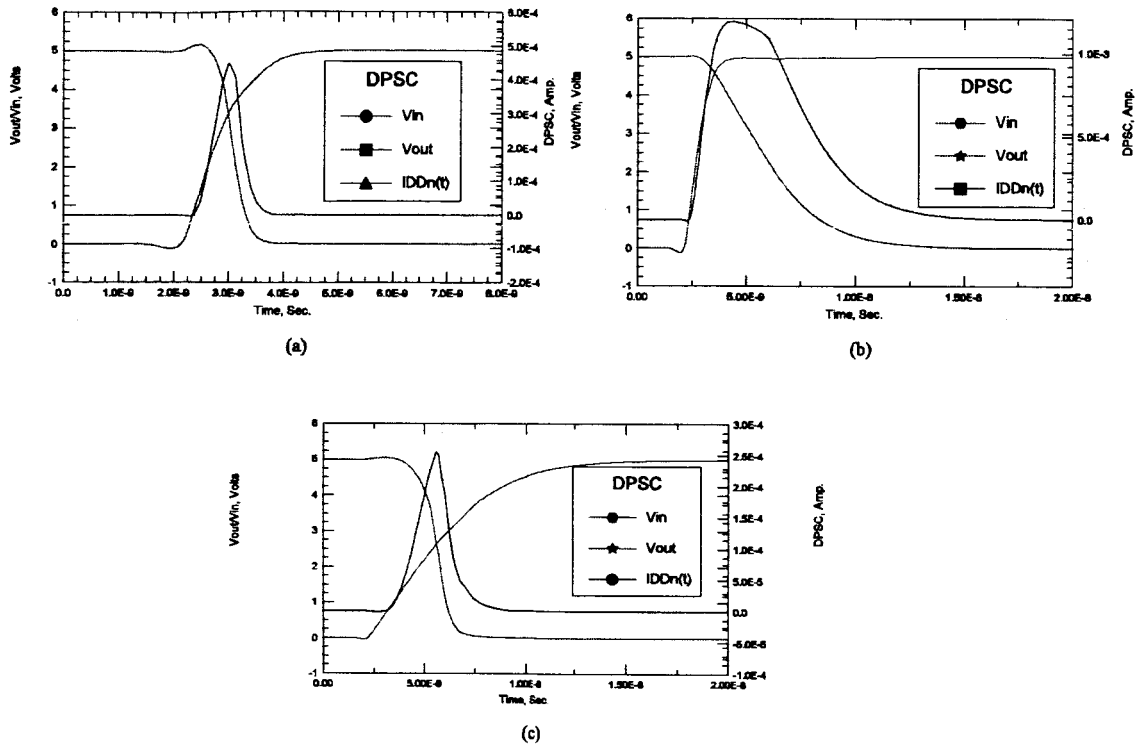


Figure 4 Various input and Output Combinations: (a) comparable input and output rise and fall times, (b) fast input, slow output, (c) slow input, fast output

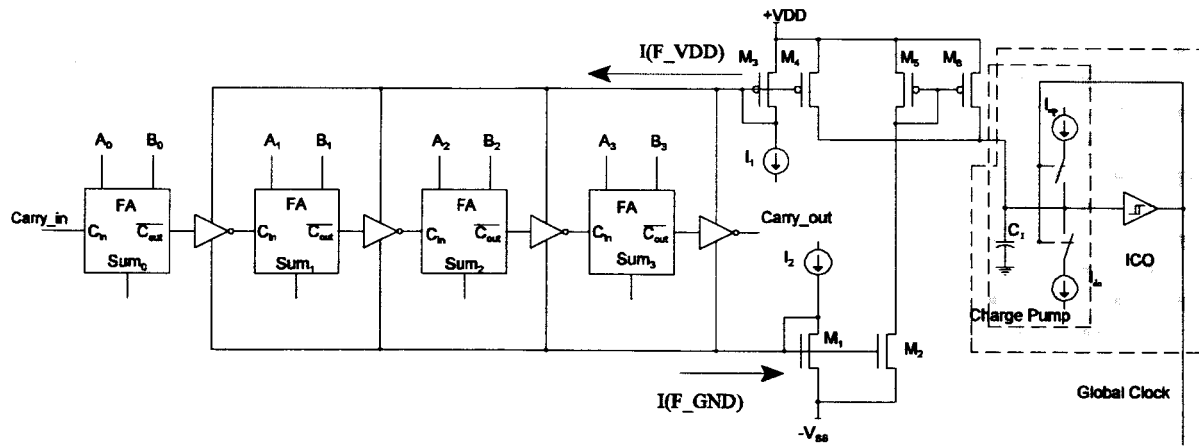


Figure 5 4-Bit Adder, DPSC monitor and ICO realizations

The relationship between the DPSC and the gate input and output transitions is shown in Fig. 3 for three input and output combinations: (a) fast input, slow output, (b) slow input, fast output and (c) comparable input fall and output rise time. This figures shows that the DPSC is non-zero during a transition at the output of a gate irrespective of the gate input/output rise and fall time characteristics. Hence, the DPSC can be used to detect transitions at circuit output(s) without regards to active path circuit parmeters.

3. Example

The DPSC due to its relationship to gate output transitions can be used to determine if the circuit being monitored, has reached steady-state. Therefore, it can be used to indicate that the output of the circuit is valid, i.e. realize a self-timed circuit. For example, if an four-bit ripple-carry adder is implemented as shown in Fig. 4, and the output inverters of the carry generation circuit, power supply and ground rail currents are summed

the net DPSC pulse width varies with the number of generated carry bits as shown in Fig. 5. The variations in the clock period and the DPSC is shown in Fig. 6 for the four-bit adder inputs A , B and $Carry$. These inputs are shown at the bottom of Fig. 5 with the corresponding clock period. The off time of the clock, i.e. the time the logic between the interstage latches is generating its output, is increased if a carry is generated by any of

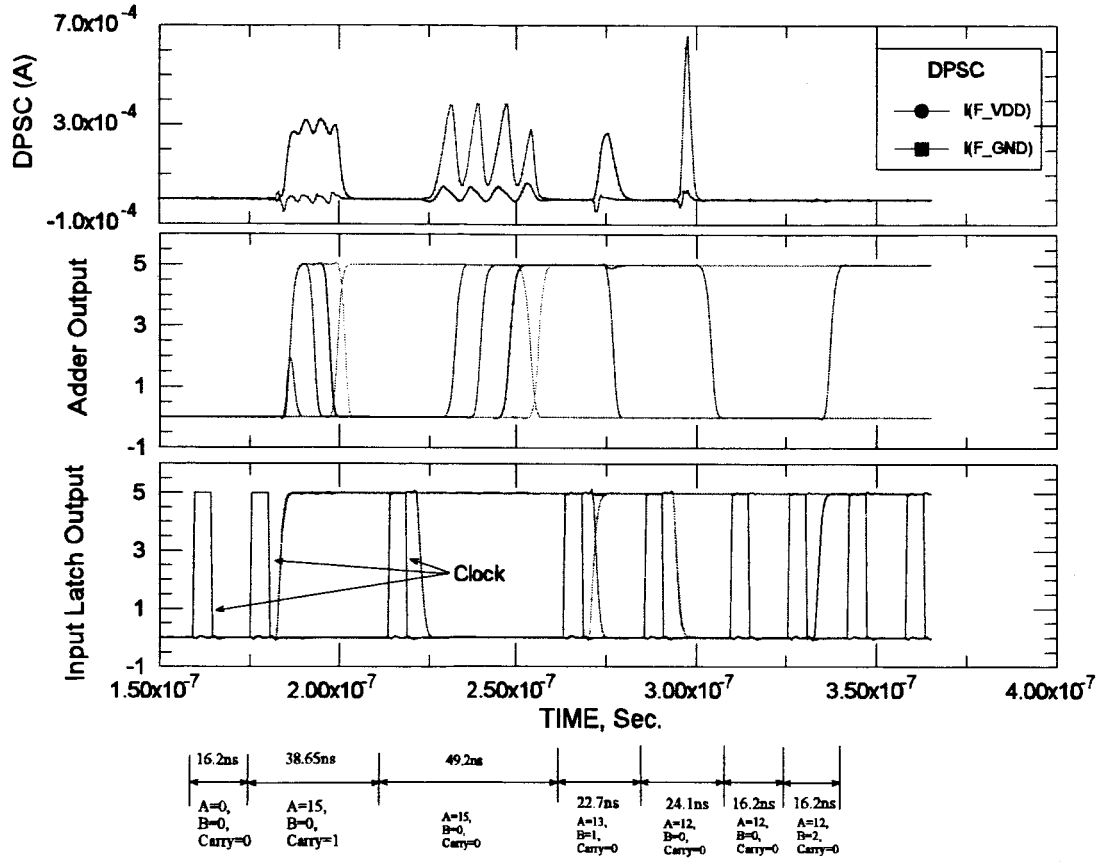


Figure 6: Adder Voltages

the full adders (i.e. FA in Fig. 4). This is done by subtracting the weighted DPSC, shown in Fig. 5, from the charge pump current source that controls the off time of the clock. This reduction in the net current entering the capacitor C_1 increase the off time. The clock off time in Fig. 4 ranges from 10.2ns to 43.2 ns (i.e. clock period - 6 nanosecond). The change in the steady-state clock period, i.e. 16.2 ns, is determined by the area under the net DPSC. This is related to circuit timing if the DPSC of the gate along the circuits' critical is

monitored. The off

time of the ICO in Fig. 4 is

$$T_{low-time} = \frac{C_1 V_{th} + \int (I(F_{VDD})(t) + I(F_{GND})(t)) dt}{I_{Down}} \quad (1)$$

where,

V_{th} - Schmitt trigger threshold voltage,

$I(F_{VDD})(t) + I(F_{GND})(t)$ - net DPSC,

I_{Down} - ICO constant current source that determines steady-state off-time.

4. Conclusion

Static CMOS circuit DPSC monitoring can be used to implement synchronous self-timed circuits if the DPSC of the gates on the critical path are monitored. This approach works because the peak value, pulse width and event duration of the DPSC is determined by the number of transitions and their appearance in time on the longest active signal path, i.e. the active subset of the critical path. This technique can be used to determine when the output of the circuit reaches steady-state because transitions occur last on the critical path. The variations in the clock period (i.e. down time) is determine by the area under the DPSC which is related to the three latter characteristics of the DPSC. The performance, i.e. the number of operands produced per unit time by a functional unit, approaches that of an asynchronous self-timed circuit, and the hardware overhead of this approach is substantially lower than asynchronous realizations.